OIL

ontology inference and interchange

On-To-Knowledge project

Topics for this presentation

- OIL as a **common core language** (4)
- aspects of OIL (5)
- OIL on the Web
  - Why XML is not enough (3)
  - Why RDF(S) is not enough (9)
  - **OIL as RDF(S) extension** (3)
- Current results & plans (4)

Requirements for a common core Ontology-language

- Well defined **syntax** (pretty obvious)
  - read ontologies
- Well defined **semantics**
  - often overlooked but equally important
  - process ("understand") ontologies
- **Expressive** enough
  - to capture many ontologies
- Easy **mapping**
  - to/from other ontology languages
- Efficient **reasoning support**

Why Reasoning Support?

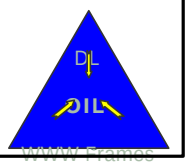
- Reasoning support is key feature of OIL-core
- Important
  - as design support tool
  - for large ontologies
  - with multiple authors
  - for integrating and sharing ontologies
- Because it allows to
  - Establish inter-ontology relationships
  - Check for consistency
  - Check for (unexpected) implied relationships
- Shown useful for DB schema integration

Ingredients for a common core

- **Frame Based Languages**
  - intuitive for many users
  - Extensive set of modelling primitives
  - OKBC, OKBC-lite, XOL
- **(Description) Logic-based languages**
  - negation and disjunction (e.g disjointness)
  - properties for slots/relations
    - e.g. transitivity for **`contained-in`**
  - Formal semantics
  - Reasoning support
    - inconsistency-detection, implicit superclass-detection

Proposed common core: OIL

- Based on standard frame languages (OKBC)
  - restricts & extends
- Has both XML and **RDF(S)** based syntax
- formalised by DL style logical constructs
- Still has frame “look and feel”
- Can still function as a basic frame language
- OIL-lite language restricted:
  - to allow for reasoning support



OIL-lite:
Restricts Frame Languages

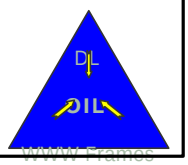
- No defaults
- limited axioms/rules
- only definition of ontology (not individuals)

Main reasons for this:

- Reasoning support
- Semantics

OIL: Extends Frame Languages

- Classes can be primitive (nec. conditions)
  - elephant $\Rightarrow$ animal that has-colour grey
- or defined (nec. and sufficient conditions)
  - vegetarian $\Leftrightarrow$ person who eats meat nor fish
- Classes allowed in slot constraints
  - slot-constraint eats has-value meat
    (eats some meat)
  - slot-constraint eats value-type meat
    (eats only meat)

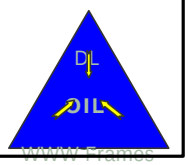


OIL: Extends Frame Languages

- Can use arbitrary class expressions instead of only class names
  - slot-constraint eats value-type NOT (OR meat fish)
- Cardinality constraints can include value-types
  - slot-constraint eats max-cardinality 1 plant
- Supports sub-slot relation
  - daughter-of sub-slot of child-of
- Slot properties
  - transitive (e.g., part-of)
  - symmetrical (e.g., connected-to)

OIL has a Formal semantics

- Defined by mapping to very expressive DL
  - slot-constraint eats has-value meat, fish
    =
    $ eats:meat Ç $ eats:fish
- Mapping is used to provide reasoning support from a DL system (e.g., FaCT)



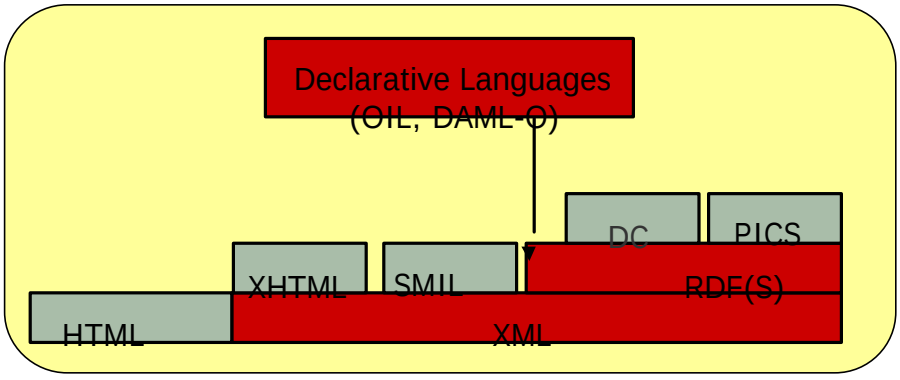
OIL (explained by example)

```
class-def animal                          % animals are a class
class-def plant                           % plants are a class
   subclass-of NOT animal                 % that is disjoint from animals
class-def tree
   subclass-of plant                      % trees are a type of plants
class-def branch
   slot-constraint is-part-of             % branches are parts of some tree
        has-value tree
        max-cardinality 1
class-def defined carnivore               % carnivores are animals
   subclass-of animal
   slot-constraint eats                   % that eat any other animals
        value-type animal
class-def defined herbivore               % herbivores are animals
   subclass-of animal, NOT carnivore      % that are not carnivores, and
   slot-constraint eats                   % they eat plants or parts of plants
        value-type plant OR (slot-constraint is-part-of has-value plant)
```

11

How to put ontologies on the Web (internet, intranet, extranet)

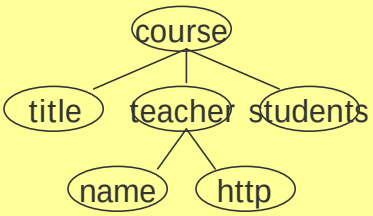
The W3C hierarchy of languages:



XML: Document = labelled tree

- node = label + attr/values + contents

```
<course date="...">
      <title>...</title>
      <teacher>...</teacher>
            <name>...</name>
            <http>...</http>
      <students>...</students>
</course>
```



- **DTD**: simple grammars to describe legal trees

- So:

why not use XML to represent ontologies?

XML: limitations for semantic markup

XML makes no commitment on:

❶ Domain specific ontological **vocabulary**

❷ Ontological **modelling primitives**

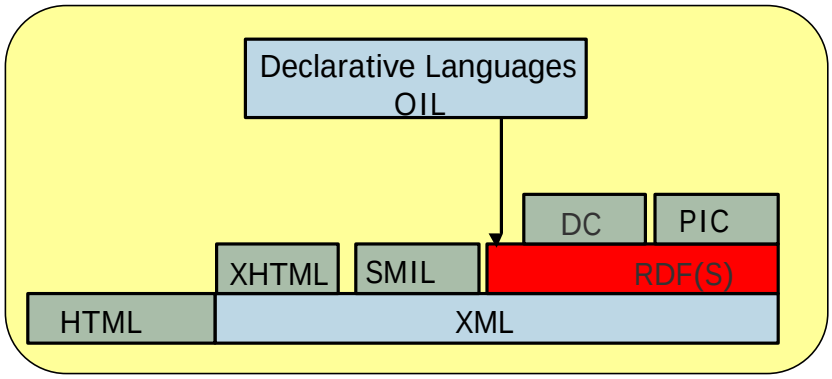
⇨ requires pre-arranged agreement on ❶ & ❷

Only feasible for closed collaboration

- agents in a small & stable community
- pages on a small & stable intranet

not for sharable Web-resources

Remember the W3C vision



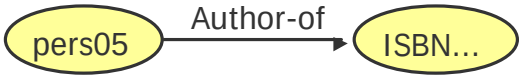
RDF(S): general background

- Intended for representation “meta-data”, basis for Web-based ontology-language
- W3C recommendation

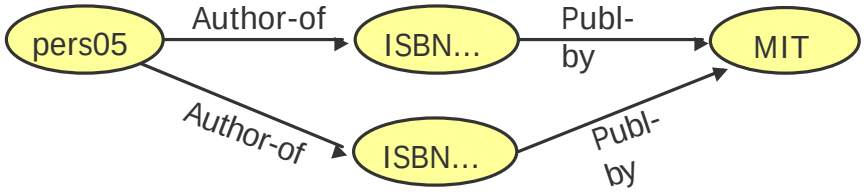
- “Because it’s there”:
  - pushed hard by W3C (TBL Himself)
  - basis of $ 80M DAML program
  - Already embraced by some vendors (Netscape)

Bluffer's guide to RDF (1)

- Object --Attribute-> Value triples



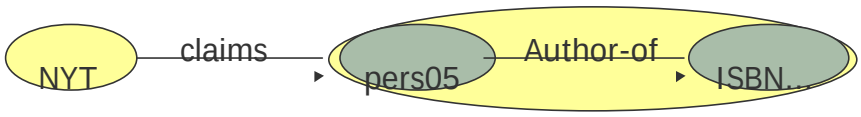
- objects are web-resources
- Value is again an Object:
  - triples can be linked
  - data-model = graph



Bluffer's guide to RDF (2)

- Object --Attribute-> Value triples
  - objects are web-resources
  - triples can be linked
  - data-model = graph

- Any statement can be an object
  - graphs can be nested



That's all there is to it

Bluffer's guide to RDF Schema

- So, RDF :
  - (very small) commitment to modelling primitives
  - but: no commitment to domain vocabulary

⇨ **RDF Schema**

- Define vocabulary for RDF
- Organise this vocabulary in a typed hierarchy
  - **`Class, SubClassOf, type`**
  - **`Property, subPropertyOf,`**
  - **`domain, range`**

That's all there is to it....

RDF Schema syntax in XML

```
<rdf:Description ID="MotorVehicle">
   <rdf:type resource="http://www.w3.org/...#Class"/>
   <rdfs:subClassOf rdf:resource="http://www.w3.org/...#Resource"/>
  </rdf:Description>

<rdf:Description ID="Truck">
   <rdf:type resource="http://www.w3.org/...#Class"/>
   <rdfs:subClassOf rdf:resource="#MotorVehicle"/>
</rdf:Description>

<rdf:Description ID="registeredTo">
   <rdf:type resource="http://www.w3.org/...#Property"/>
   <rdfs:domain rdf:resource="#MotorVehicle"/>
   <rdfs:range rdf:resource="#Person"/>
</rdf:Description>

<rdf:Description ID="ownedBy">
   <rdf:type resource="http://www.w3.org/...#Property"/>
   <rdfs:subPropertyOf rdf:resource="#registeredTo"/>
</rdf:Description>
```

State-of-the-art@W3C

- RDF: to represent "meta-data"
- RDF-S: to define vocabulary for RDF

- RDF is data-model + syntax
  - only a very weak semantic interpretation
  - no inference model
- RDF-S goes a step further, but still
  - **no** precisely described meaning
  - **no** inference model

Quote from Ora Lassila (RDF)

Future: We Need More!

- Structural modeling obviously not enough
  - we need a "logic layer" on top of RDF
  - some type of description logic is a possibility (after all, we are talking about frame systems)
- Exposing a wide variety of data sources as RDF is useful, particularly if we have logic/rules which allow us to draw inference from this data
- My proposal: RDF + DL = "Frame System for WWW"
  - this is probably a good starting point for DAML as well (details to be worked out by this workshop)

Quote from Henry Thompson

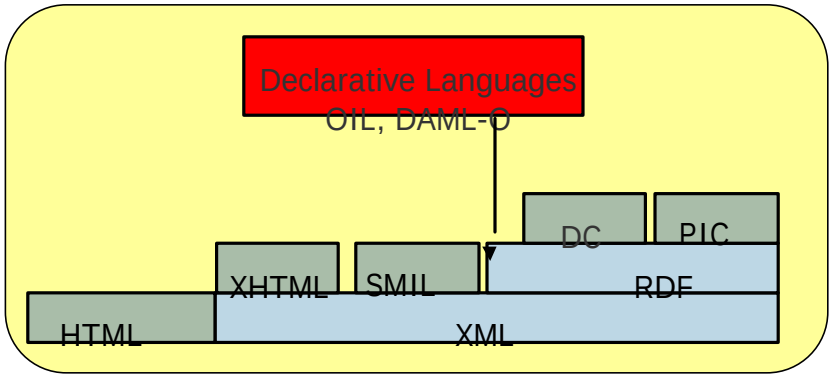
"The Semantic Web needs **a** logic on top"

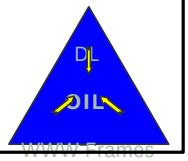
NB: "a" logic = the box with the crank
1 FOL

OIL =

modelling primitives from **frames** *(OKBC-lite)*
+ semantics and inference from **Description Logic**
+ syntax from **RDF(S) & XML(S)**

OIL for the Semantic Web





OIL as RDF(S) extension (1/2)

```
<rdfs:Class rdf:ID="herbivore">
  <rdf:type
      rdf:resource="http://www.ontoknowledge.org/#DefinedClass"/>
  <rdfs:subClassOf rdf:resource="#animal"/>
  <rdfs:subClassOf>
    <oil:NOT>
      <oil:hasOperand rdf:resource="#carnivore"/>
    </oil:NOT>
  </rdfs:subClassOf>
</rdfs:Class>
```


OIL as RDF(S) extension (1/2)

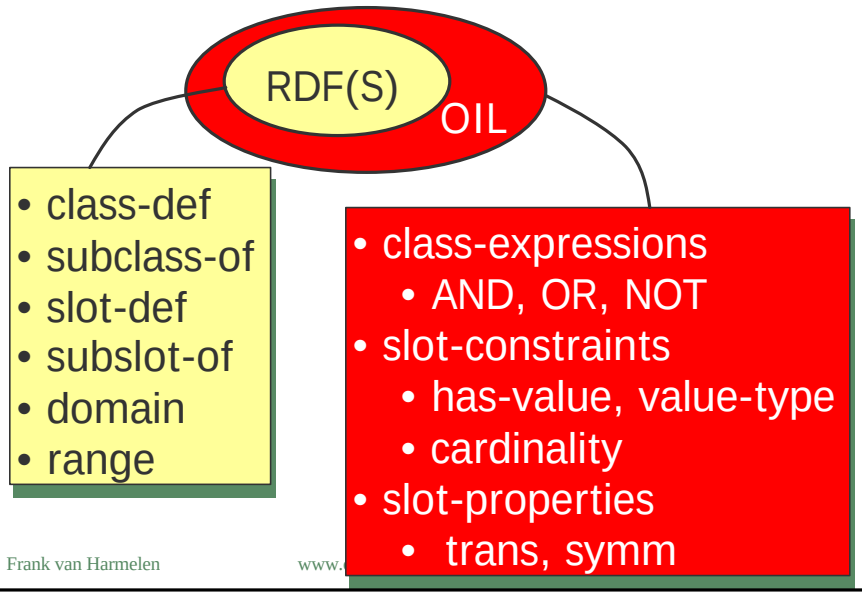
```
<rdfs:Class rdf:ID="herbivore">


  <rdfs:subClassOf rdf:resource="#animal"/>
  <rdfs:subClassOf>



  </rdfs:subClassOf>
</rdfs:Class>
```

OIL as RDF(S) extension (2/2)



OIL as the basis for DAML-O

- DAML = $80M DARPA program for Semantic Web
- Ontologies are regarded as fundamental
- first version of DAML-O out < end 2000
- mandatory use for all DAML participants (W3C, Stanford, ISI, Lockheed, MIT, Nokia,…)
- OIL-lite and Standard OIL are the most likely basis for DAML-O

OIL: currently available tools

- Definition of language
  - semantics
  - XML encoding
  - RDF encoding
- Tools:
  - translators (XSL based)
  - reasoner (FaCT, DL-based)
  - OntoEdit
- case-studies
  - GIS ontology mapping
  - $(KA)^2$ ontology
  - CIA world fact book

OIL: some collaborating parties

- EU academics
  - University of Bremen
  - Univ. of A'dam
  - OU-UK
  - Univ. Manchester
  - A'dam Medical Centre
- EU IST Projects
  - On-To-Knowledge
  - IBROW
  - Comma
- US academics:
  - Univ. of Stanford (DB, KSL, Med.Inf)
  - Univ. of Maryland
  - SRI
- outside academia
  - W3C (RDF Working group)
  - DARPA (DAML initiative)
- EU industrials:
  - Swiss Life
  - BT
  - CognIT
  - AIdministrator

OIL: current & future work

- Layered approach to anguage extensions
  RDF(S) ⊂ OIL-lite ⊂ Standard OIL ⊂ OIL layer 1 ⊂…
  - axioms, concrete domains, modules, defaults,…
- Ontology construction
- Ontology evolution
- Ontology mapping